

GSoC'23: Python Software Foundation

PyAr: Python Argentina

**PyZombis**



## About me:

Name : Soma Sujith

Time Zone: IST (GMT +5:30)

Address : Delhi, India

GitHub : <https://github.com/sujith-rek>

## Background

I'm a second-year undergraduate pursuing B.Tech in Computer Science at Shiv Nadar University. I completed courses like Object Oriented Programming, Data structures and algorithms, and Operating Systems as part of my studies. I'm familiar and worked with frameworks like ReactJS, NextJS, NodeJS, HTML, FastAPI, and Django. Used python in my day-to-day use for academic projects. I can quickly adapt to new technologies and learn the necessary.

# Contributions to PyZombis

- [PoC for PyScript](#) PyScript which can be embedded into HTML to run python
- [PoC](#) for Pygame Challenge
- [TWP47](#) Changed the static code which isn't executable to executable. Added E2E tests
- [TWP30](#) Simplified Random number exercises which solved issue [#210](#) and added E2E tests
- [TWP40](#) Adding modulo division code for the lecture
- Enabled [Code lens](#) for the required activities

## Proposal

### Project Abstract: PyZombis

PyZombis is a community course to teach Python to the Spanish-speaking community. It is based on a Brazilian MOOC Python para Zumbis. The idea is to make an improved version of the course in Spanish, with an interactive environment for the students, that allows them to visualize and try active code on the site. Thanks to the previous GSoC contributor, the project now has SQL and Pygame wrappers for Brython. I plan to use this to add and improve lectures for the MOOC.

### Milestones

Phase - 0 ( Community bonding period)

During the community bonding period, I'd to discuss the goals to be achieved in the period and set down appropriate deadlines, and deliverables. Get to know and understand the requirements needed to complete the project.

## 1. Saving of exercise progress using JS [Local storage](#)

For this we can take the code from active code & codelens and then store it in the local storage, when the student visits the page later, using local storage we can restore the progress. This can be done using the Web APIs provided by every browser. An appropriate script will be written to get this done.

## 2. Improve PyGame Lectures

The current pygame lectures lack appropriate explanations and only contain code. To improve them, I plan to use the existing zombie game as an example to teach various actions, keyboard events, and other related concepts. If using the zombie game is not feasible, I will create additional content to teach how to create a game step-by-step.

[PoC](#) can be found here,

I'd like to see what else could be added like we load an audio file and asking students to include that in-game, etc.,

Using the current Zombis game

- Creating a player object
- How to add music, render images, setting up fonts(banners)
- Keyboard events (Keyup,keydown)
- Can modify and write pygame challenge which guides and makes the student to do a classic snake game

## 3. [PyScript](#)

HTML-based IDE, which runs python online.

By the end of this phase, a Web IDE can be added to the project where students can practice python generally. We can also add the second IDE where students can work with visualization modules separately

[Packages/modules](#) supported by Pydiode which runs the pyscript

Currently pygame isn't supported.

Available lectures don't explain - different types of datatypes available in python and data structures, Object-oriented programming topics

We can use the currently made Pyscript Code runner. To teach these,

typically we can add the topics like List, Tuple, Array, Set, Stacks, Classes, Objects, Polymorphism, Encapsulation, Inheritance, and Data Abstraction.

## Stretch goals

If I complete the previous phases early than expected, I'll try to solve the pending issues, establish baselines to improve the project, and do my best to improve the project.

## Timeline

Period	Goal for the Period
Community bonding May 4 to May 28	<ul style="list-style-type: none"><li>• Complete all the goals of Phase 0</li></ul>
Week 1&2 May 29 to June 12	<ul style="list-style-type: none"><li>• Work on the ways to hook/get the code from active code</li></ul>
Week 3&4 June 13 to June 26	<ul style="list-style-type: none"><li>• Implement the local storage for the project</li></ul>
Week 5&6 June 27 to July 10	<ul style="list-style-type: none"><li>• Start implementing the additional pygame</li></ul>
Mid Term Evaluation July 14	<ul style="list-style-type: none"><li>• UP and running local storage feature</li></ul>
Week 7&8 July 14 to July 28	<ul style="list-style-type: none"><li>• Complete the lectures</li><li>• Add appropriate tests for the same</li></ul>
Week 9&10 July 29 to August 11	<ul style="list-style-type: none"><li>• Enhance current made Pyscript IDE and start implementing the lectures</li></ul>
Week 11 August 12 to August 18	<ul style="list-style-type: none"><li>• Complete the PyScript task</li></ul>
Week 12 August 19 to August 28	<ul style="list-style-type: none"><li>• Complete any Pending work, documentation, etc...,</li></ul>

NOTE: This timeline is proposed/draft actual timeline of the project tend to vary a bit and will we adjusted accordingly

## Other Commitments

I don't have any other commitments apart from GSoC for this summer. My summer break starts from mid-May until mid-August, i.e., Week 12 of GSoC, and this doesn't affect my contributions for that week.

I can consistently contribute between 40-45 hours per week throughout the Project duration.

## Post GSoC'23

Easy/Simple setup for any open-source contribution is essential. PyZombis use many libraries, and all of them are interdependent. Initially, when setting up the project, I encountered many library issues. Like some components aren't working for the latest versions of python. For instance, the current version of runestone is 6.5.4, but the running version for the project is 5.8.1 with python3.8/3.9.

After my GSoC, I want to work in this area and improve the project. I'd love to help the new contributors to get started with PyZombis.